

# Metode *Bayesian Networks* untuk Menyelesaikan *Occlusion* pada *Object Tracking*

Ikhwan B. Zarkasi, Rose Maulidiyatul H, Dina Nurul Fitria\*

Program Studi Teknik Informatika, Fakultas Sains dan Teknologi,  
Universitas Al Azhar Indonesia, Jl. Sisingamangaraja, Jakarta 12110

\*Penulis untuk Korespondensi: [dina.fitria@uai.ac.id](mailto:dina.fitria@uai.ac.id)

**Abstrak** – Banyak cara untuk dapat mendeteksi keamanan sebuah wilayah tertentu. Salah satu cara pengamanan yang bisa digunakan adalah dengan menggunakan pemantauan berbasis video pengawasan (*video surveillance*). Sebenarnya, video pengawasan sudah banyak digunakan di Indonesia. Tetapi, umumnya video pengawasan ini hanya mampu merekam gambar, tanpa ada kemampuan pintar yakni, *object tracking*, *object recognition* dan *object analyzing*. Sehingga, hasil yang diharapkan kurang maksimal dan belum bisa membantu tugas pengawasan secara keseluruhan. Paper ini bertujuan untuk membuat algoritma dari *object tracking* yang ada pada video pengawasan sebagai rujukan pengembangan video pengawasan dengan kemampuan *object recognition* dan *object analyzing*. Masalah utama yang sering muncul dalam pembuatan *object tracking* adalah ketika terjadi *occlusion* (tumpang tindih) antara dua *object* dalam sebuah frame. Pada saat *occlusion*, *object* yang sama pada frame yang berbeda kemungkinan dapat dikenali sebagai *object* yang berbeda. Sehingga, proses *object tracking* akan menjadi terganggu. *Bayesian Networks* memungkinkan untuk membandingkan data yang didapat dari masing-masing *object* yang ada (*likelihood*) dengan data awal yang telah dimiliki (*prior*), dengan menghitung *Maximum A-Posteriori Probability*(MAP) yang dimiliki, sehingga *object* yang sama pada frame yang berbeda tetap akan dikenali sebagai *object* yang sama

**Abstract** – There are many ways/technique to detect the security/safety of fixed area. One of security technique that can be used is by using monitoring based on Video surveillance. In fact, this monitoring video has already been used in Indonesia. But, video surveillance, commonly,

just can record images without any smart abilities, such as *object tracking*, *object recognition* and *object analyzing*. So, the expected result is not optimal and still not be able to help monitoring role totally. This research is aimed to make the algorithm of *object tracking* in video surveillance, in order to be reference for development of video surveillance with ability of *object recognition* and *object analyzing*. The main problem that frequently comes up on the making of *object tracking* is *occlusion* between two objects in a single frame. When *occlusion* is happened, same object in different frame probably can be recognized as two different objects. So, the process of *object tracking* can be disturbed. *Bayesian Network* is enable to compare data that got from every object (*likelihood*) with prior data that has already been provided by counting its *Maximum A-Posteriori Probability* (MAP), so same object in different frame are still be able to be recognized as same object.

**Keywords** - *Bayesian Network*, *occlusion*, *object tracking*, *algoritma*, *Maximum A-Posteriori Probability* (MAP).

## I. PENDAHULUAN

Pemantauan berbasis video pengawasan banyak digunakan di Indonesia untuk pengawasan keamanan ruang publik. Tetapi, umumnya video pengawasan ini hanya mampu merekam gambar tanpa memiliki kemampuan pintar. Selain itu, video pengawasan ini membutuhkan operator untuk mengawasi video, sehingga akan meningkatkan resiko terjadinya *human error* yang dikarenakan kelelahan, kejenuhan dan berbagai faktor yang lain.

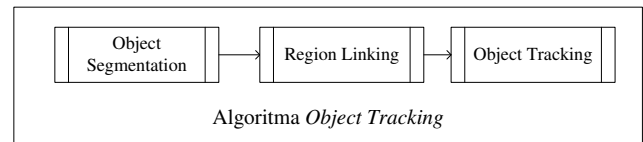
Tugas pengawasan ini akan terbantu jika video pengawasan memiliki kemampuan untuk dapat mendeteksi dan mengenali benda (*object*) yang berada dalam ruang terbatas, baik itu posisi ataupun kegiatan yang sedang dilakukan oleh *object* tersebut. Dengan kemampuan ini, operator tidak perlu ada didepan video pengawasan sepanjang waktu. Untuk dapat membuat video pengawasan dengan kemampuan tersebut, maka perlu dipecahkan permasalahan yang mungkin muncul pada saat pendeteksian dan pengenalan benda seperti *object tracking*.

Masalah yang sering ditemukan dalam pembuatan algoritma *object tracking* adalah adanya *occlusion* antara *object* yang berbeda. Terdapat beberapa metode yang dapat digunakan untuk menyelesaikan permasalahan *occlusion*, seperti *HOG*, *Expectation Maximization Algorithm*, *Hybrid Strategy*, *Online Adaptive Neural Classifier* atau *Bayesian Network*<sup>[1]</sup>. Semua metode tersebut menggunakan kesamaan karakteristik *object* dalam menyelesaikan permasalahan *occlusion*, tanpa memperdulikan pengetahuan awal yang ada, berdasarkan percobaan yang telah dilakukan sebelumnya. Tetapi pada *Bayesian Network*, pengetahuan awal tersebut digunakan, selain penggunaan kesamaan karakteristik untuk menyelesaikan permasalahan *occlusion*. Dengan menggunakan pengetahuan awal ini maka dimungkinkan hasil yang didapatkan untuk pemilihan *object* menjadi lebih *robust*. Karena itu, paper ini menggunakan metode *Bayesian Network* untuk penyelesaian *occlusion*.

Terdapat beberapa paper ilmiah yang mengulas tentang penggunaan *Bayesian Network* untuk penyelesaian masalah *occlusion*<sup>[2,3,4,5]</sup>. Paper ilmiah tersebut digunakan sebagai rujukan dalam pembuatan paper ini. Dalam paper ini, pengenalan terhadap *object* yang sama hanya dilakukan terhadap *object* sebelum dan sesudah *occlusion*. Untuk *object* yang merupakan hasil *occlusion*, akan dikenali sebagai *object* baru. Tugas akhir ini hanya membahas penyelesaian *occlusion* menggunakan *Bayesian Network* tanpa melakukan perbandingan lebih lanjut dengan metode yang lain.

## II. TINJAUAN PUSTAKA

Perancangan algoritma *object tracking* pada paper ini terbagi menjadi 3 modul besar yakni, *object segmentation*, *regionlinking* dan *object tracking*.



Gambar 1. Algoritma global *object tracking*

Modul *object segmentation* digunakan untuk memisahkan atau mengisolasi *object* yang bergerak dari *background* yang ada. Sehingga akan didapatkan masing-masing *object* dalam 1 frame.

Modul *region linking* digunakan untuk menghubungkan masing-masing *object* yang sama dalam frame berbeda, sehingga akan terbentuk hubungan antar *object* yang sama yang digunakan untuk proses selanjutnya.

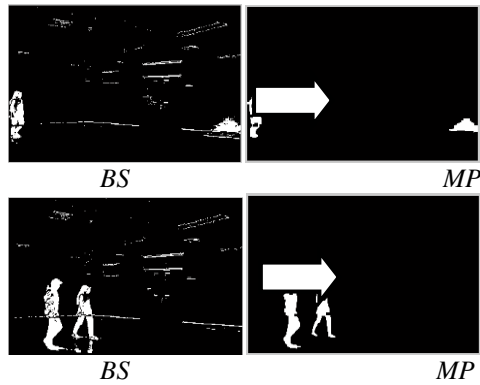
Modul *object tracking* digunakan untuk memberikan label kepada *object* yang sama dalam frame berbeda. Walaupun *object* tadi merupakan *object* hasil dari *occlusion*.

Masing-masing modul tersebut akan diulas dalam bab pembahasan.

## III. METODOLOGI PENELITIAN

Modul *object segmentation* ini menggunakan algoritma *background subtraction (BS)* untuk memisahkan *object* yang diobservasi dari *background* yang ada. Karena hasil dari *background subtraction* biasanya masih banyak mengandung *noise* atau derau, maka dilakukan *morphological process (MP)* sebagai proses lanjutan. Tujuan dari *morphological process* ini adalah sebagai filter untuk membersihkan frame dari *noise*.

*Morphological process* yang digunakan tergantung dari seberapa besar serta bentuk dari *noise* yang ada. Cara yang umum digunakan bisa menggunakan *dilation*, *erotion*, *closing*, *opening* atau *median filter*. Contoh hasil dari *object segmentation* ini adalah :



Gambar 2. Hasil Segmentasi Frame ke-49 dan 125 pada percobaan video pertama

#### IV. HASIL DAN PEMBAHASAN

*Region linking* ini digunakan untuk menghubungkan *object* yang sama dalam frame berbeda. *Object* yang telah didapatkan dari modul *object segmentation* akan dibandingkan karakteristiknya dengan *object* pada frame sebelumnya. Apabila terdapat perbedaan yang kecil (masih dalam batas *threshold* yang ditentukan) dalam karakteristiknya, maka kedua *object* tersebut akan dibuat suatu hubungan.

Karakteristik yang digunakan disini adalah warna, area dan centroid. Untuk menghitung besar perbedaan karakteristiknya, digunakan persamaan *Euclidean distance* :

$$d_{ij} = \| \mathbf{x}_i - \mathbf{y}_j \| = \sqrt{(x_{1i} - y_{1j})^2 + \dots + (x_{pi} - y_{pj})^2} \quad (1)$$

dengan  $\mathbf{x}_{(ij)}$  adalah *object* pada *current frame* (frame yang diamati), dan  $\mathbf{y}_{(ij)}$  adalah *object* yang ada pada frame sebelumnya atau *reference frame*. Hasil dari modul ini adalah *trajectory* atau hubungan antar *object* yang sama dalam frame berbeda yang tersimpan dalam suatu matriks. Disini, juga dilakukan pengendalian terhadap *noise* yang tidak terhapus dalam modul *object segmentation*. Suatu *object* akan dianggap sebagai *noise* jika *trajectory* yang dihasilkan tidak terlalu panjang.

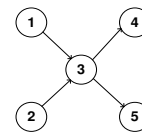
Modul *object tracking* ini berfungsi untuk memberikan label terhadap masing-masing *object* yang terdeteksi. Untuk *object* yang sama akan diberikan label yang sama meskipun berada dalam frame yang berbeda. Masukan dari modul ini adalah matriks hasil dari *region linking*. Matriks ini menggambarkan hubungan antara *object* yang sama dalam frame yang berbeda. Apabila terjadi

*occlusion*, maka matriks yang dihasilkan akan menjadi rusak sehingga dibutuhkan modul *object tracking* untuk memberikan label yang tepat. Modul ini terbagi menjadi 4 tahap :

##### 4.1 Make Graphical Model

Pada tahap ini akan dibuat hubungan antar *object* yang berbeda. *Object* yang berbeda didapatkan dari *trajectory* yang dihasilkan modul *region linking*. Untuk memudahkan, satu *trajectory* menggambarkan sebuah *object*, dan sebuah *object* diwakili oleh sebuah *node*. Semua *node* ini akan dihubungkan, sehingga akan terbentuk hubungan *parent* dan *child* antar *node*.

Pada paper ini diberikan batasan maksimal jumlah *parent* dan *node* sebesar  $2^{[4]}$ .



Gambar 3. Contoh *graphical model* percobaan video pertama

##### 4.2 Make Admissible Label

Setelah didapatkan *graphical model* dari masing-masing *node*, maka akan dicari kemungkinan label yang dimiliki oleh sebuah *node*, kemungkinan label ini merupakan gabungan dari label *parent* dari *node* yang diamati dengan label baru yang diberikan kepada *node* yang diamati.

$$L_i = \left[ \bigcup_{j \in I_i} L_j \right] \cup L_{new} \quad (2)$$

Dimana :  $L_i$  : kemungkinan label *node* yang sedang diamati

$L_j$  : gabungan kemungkinan label semua *node parent* dari *node* yang sedang diamati

$L_{new}$  : label baru yang dimiliki oleh *node* yang sedang diamati

##### 4.3 Choose Label

Dari kemungkinan label yang didapatkan, maka akan dilakukan pemilihan, label mana yang paling cocok untuk *node* yang diamati. Pemilihan ini dilakukan dengan menghitung *Bayesian Networks* yang dimiliki oleh masing-masing *node* yang diamati. Persamaan *Bayesian Networks* yang digunakan :

$$P(A_i|B) = \frac{P(B|A_i)P(A)}{\sum P(B|A_i)P(A)} \longrightarrow \boxed{f(\theta|X) = \frac{f(X|\theta)f(\theta)}{f(X)}}$$

karena besar pembagi (*evidence*) selalu sama untuk masing-masing *node* dalam satu percobaan, maka persamaan diatas bisa ditulis :

$$\boxed{f(\theta|X) \propto f(X|\theta)f(\theta)} \quad (3)$$

Persamaan tersebut merupakan perkalian dari *likelihood* dan *prior* yang dimiliki oleh masing-masing *node*. *Prior*, merupakan data subjektif awal yang dimiliki oleh masing-masing *node* berdasarkan percobaan yang telah dilakukan. Dalam paper ini, nilai *prior* ditentukan berdasarkan jumlah dari *parent* yang dimiliki oleh *node* yang sedang diamati, yakni :

$$\boxed{\text{prior} = \begin{cases} P_{\text{new}} & \text{if Parent} < 2 \\ \frac{1-P_{\text{new}}}{n} & \text{if Parent} = 2 \end{cases}} \quad (4)$$

Sedangkan, *likelihood* merupakan data percobaan yang dimiliki oleh masing-masing *node*. *Likelihood* ini bersifat random. Perhitungan *likelihood* dalam paper ini berdasarkan karakteristik yang dimiliki oleh suatu *node*. karakteristik yang digunakan ada 6 yakni :

- luas area
- kecepatan rata-rata
- arah
- rata-rata warna merah
- rata-rata warna hijau
- rata-rata warna biru.

Karakteristik yang dimiliki tersebut akan dibandingkan dengan masing-masing kemungkinan label yang dimiliki oleh *node* yang diamati. Perhitungan besar *likelihood* ini dilakukan dengan menggunakan persamaan distribusi binomial :

$$\boxed{x = \sum_{k=1}^K \binom{n}{k} p^k (1-p)^{(n-k)}} \quad (5)$$

dimana :  $K$  = jumlah kemiripan karakteristik yang dimiliki oleh *node* yang diamati dibandingkan dengan kemungkinan label yang diberikan  
 $n$  = jumlah karakteristik yang dibandingkan (untuk tugas akhir ini bernilai 3)

$P$  = probabilitas untuk kemiripan 1 buah karakteristik (probabilitas kejadian sukses)  
 $1 - P$  = probabilitas kejadian gagal

Setelah didapatkan nilai *prior* dan *likelihood* untuk masing-masing *node*, mengadopsi teori MAP (*Maximum A-posteriori Probability*), maka pemilihan label dilakukan dengan mengambil nilai maksimal dari perkalian yang dihasilkan.

$$\text{posterior} = \max(\text{likelihood} * \text{prior}) \quad (6)$$

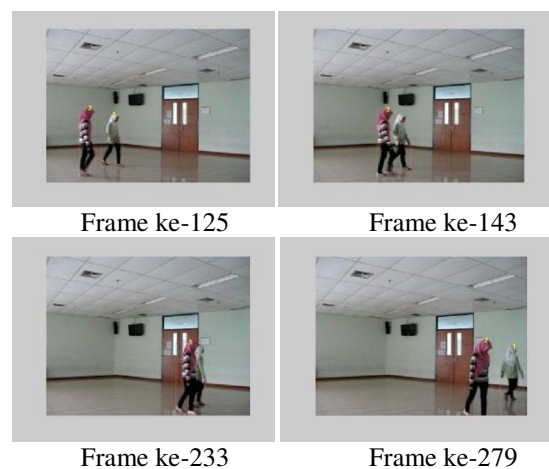
#### 4.4 Update Matrix

Matriks yang dihasilkan pada modul *region linking*, selanjutnya akan di-*update* berdasarkan label yang telah didapatkan. Sehingga, akan diperoleh label yang sama untuk *object* yang sama.

#### 4.5 Uji Coba

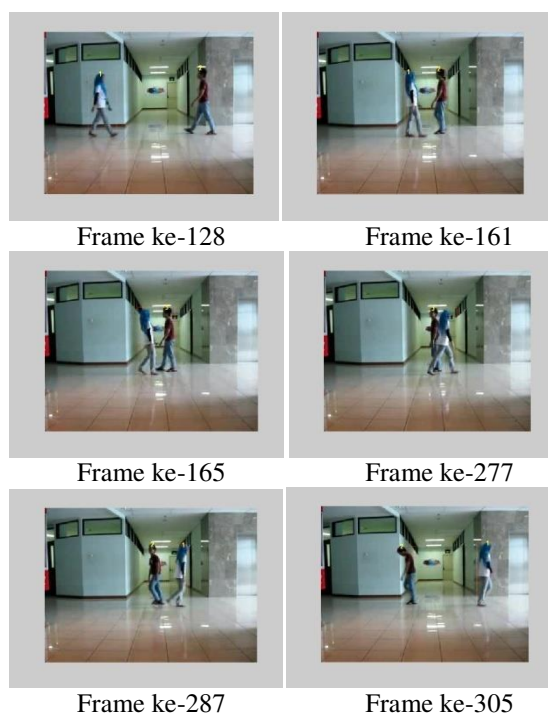
Eksperimen dilakukan dengan menggunakan 6 buah video dengan keadaan dan jumlah *object* yang berbeda. Jumlah *object* bervariasi mulai dari 2 hingga 4 *object*. Video yang digunakan diperoleh dari hasil perekaman menggunakan kamera Canon Powershot A580 dengan resolusi 320x240. Pengambilan gambar ini dilakukan di 2 tempat yakni lantai 6 serta lantai 3 Universitas Al-Azhar Indonesia, sekitar pukul 13.00 – 15.00 tanpa menggunakan lampu penerangan (hanya menggunakan sinar matahari). Tiap video yang diambil memiliki durasi berkisar antara 10 – 20 detik dan memiliki kecepatan 25fps

Percobaan Pertama:



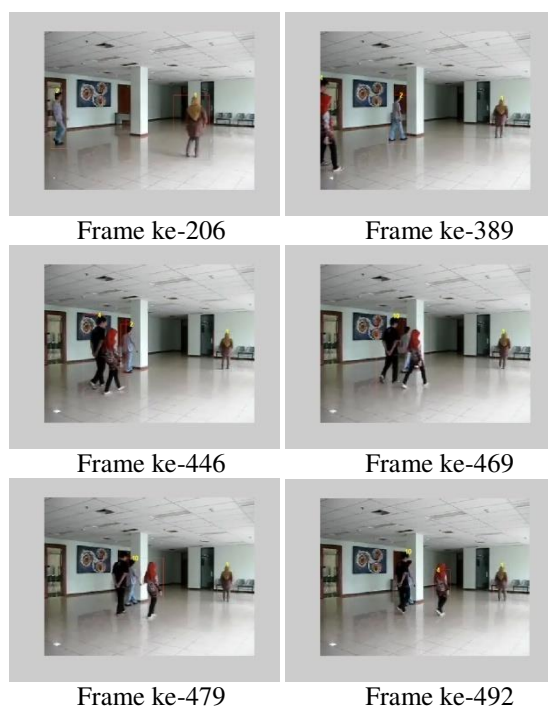
Gambar 4. Hasil *Object Tracking* pada percobaan video pertama

Percobaan ketiga:



Gambar 5. Hasil *Object Tracking* pada percobaan video ketiga

Percobaan keenam:



Gambar 6. Hasil *Object Tracking* pada percobaan video keenam

Hampir semua orang dapat dikenali dan diberikan label yang sesuai, kecuali untuk video yang keenam. Di video keenam terdapat 2 *object* yang tidak dapat dikenali sempurna, hal ini disebabkan karena, *object* yang ada sebelum *occlusion* dan *object* yang dihasilkan sesudah *occlusion* merupakan *object* yang sama sekali berbeda (gabungan dari beberapa *object*). Metode *Bayesian* dapat digunakan dengan sempurna untuk semua video (kecuali video keenam). Percobaan yang dilakukan disini menggunakan *object* yang memiliki warnabaju yang berbeda, meskipun ada beberapa *object* yang memiliki warna baju agak mirip (video keempat, kelima dan keenam).

Waktu rata-rata pemrosesan untuk setiap frame dalam video diatas berbeda satu sama lain, tergantung dari *object* yang terdeteksi dalam masing-masing video dan lama video yang diproses. Semakin banyak *object* yang terdeteksi atau semakin panjang video inputan, maka waktu yang dibutuhkan semakin lama. Untuk tugas akhir ini waktu rata-rata yang dibutuhkan untuk memproses sebuah frame (dari semua rata-rata waktu yang dibutuhkan untuk semua video) adalah 0.49 detik. Performansi ini dilakukan dengan menggunakan komputer dengan spesifikasi Processor Intel(R) Pentium(R) Dual CPU T2390 @ 1.86 GHz dan RAM sebesar 1014MB. Untuk lebih jelasnya dapat dilihat dari tabel dibawah ini:

Tabel 1. Waktu yang dibutuhkan untuk memproses video

Video	Panjang (frame)	Waktu total (dtk)	Waktu rata-rata/frame (dtk)
pertama	350	133.05	0.38
kedua	392	142.89	0.36
ketiga	357	159.23	0.45
keempat	313	146.48	0.47
kelima	420	268.961	0.64
keenam	497	304.68	0.61
Total	2329	1155.29	0.49

## V. KESIMPULAN

1. Untuk mendapatkan hasil *object tracking* yang *robust*, maka proses segmentasi harus dilakukan dengan benar. Jika segmentasi yang dihasilkan tidak sempurna, maka banyak *noise* yang akan ikut terdeteksi sehingga mengganggu proses *object tracking*.
2. Metode *Bayesian Networks*, dapat digunakan untuk menyelesaikan permasalahan *occlusion*



yang ada pada *object tracking*, dengan pengetahuan awal (*prior*) yang dimiliki.

3. Algoritma *object tracking* ini sangat dipengaruhi oleh banyak object yang terdeteksi serta panjang video masukan. Semakin banyak object yang terdeteksi atau semakin panjang video masukan, maka semakin lama waktu yang dibutuhkan untuk melakukan pemrosesan.
4. Performansi algoritma *object tracking* ini cukup bagus. Hal ini terlihat dari hasil pelabelan terhadap object yang terdeteksi dan waktu rata-rata yang dibutuhkan untuk memproses sebuah frame, yakni sebesar 0.4960 detik. Performansi ini dilakukan dengan menggunakan computer dengan spesifikasi Processor Intel(R) Pentium(R) Dual CPU T2390 @ 1.86 GHz dan RAM sebesar 1014MB.

## DAFTAR ACUAN / PUSTAKA

- [1] P. Bilinski, F. Bremond, M. Becha Kaaniche, "Multiple Object Tracking with Occlusion Using HOG Descriptors and Multi Resolution Images," *IEEE*, 2004.
- [2] A. Abrantes, J. Marques, and J. Lemos, "Long Term Tracking Using Bayesian Network," *IEEE ICIP*, III:609-612, September 2002.
- [3] M. Narayana, D. Haverkamp, "A Bayesian algorithm for tracking multiple moving objects in outdoor surveillance video," *IEEE CVPR*, pp.1-8, June 2007.
- [4] P. Jorge, J. Marques, and A. Abrantes, "Estimation of the Bayesian Network Architecture for Object Tracking in Video Sequence," *ICPR '04-vol2*, pp.732-735, August 2004.
- [5] B. Han, Y. Zhu, D. Comaniciu, and L. Davis, "Kernel Based bayesian Filtering for Object Tracking," *IEEE CVPR '05-vol1*, June 2005.
- [6] Gonzalez, Rafael C., Woods, Richard E., *Digital Image Processing Second Edition*. Prentice Hall, New Jersey, 2002.
- [7] Walpole, Ronald E., Myers, Raymond H., Myers, Sharon L., Ye, Keying., *Probability & Statistic for Engineers & Scientist Seventh Edition*. Prentice Hall, New Jersey, 2002